ISSN:2153-182X, E-ISSN: 2153-1838
Vol. 17 No. 02 (2023)

AN ENCIPHERMENT ALGORITHM FOR ENCRYPTION AND DECRYPTION IN CLOUD ENVIRONMENT

M.S. Ajitha Purnima
Research Scholar, Dr. MGR Educational and Research Institute, Chennai, INDIA

Dr. Devendran. A
Professor, Dr. MGR Educational and Research Institute, Chennai, INDIA

Dr. Rajavarman V N
Professor, Department of Computer Science and Engineering, Dr MGR Educational and Research Institute, Chennai, INDIA

Abstract

Cloud computing has emerged as a transformative technology, offering scalable and flexible solutions for data storage, processing, and application hosting. However, this paradigm shift brings forth a myriad of security issues and challenges that demand careful consideration. This paper explores the key security concerns in cloud computing, including data breaches, identity management, compliance, and shared infrastructure vulnerabilities. The dynamic nature of cloud environments introduces complexities in ensuring data confidentiality, integrity, and availability. Additionally, issues related to trust, multi-tenancy, and the shared responsibility model between cloud service providers and users further complicate the security landscape. This paper aims to provide insights into the evolving nature of security threats in the cloud, emphasizing the need for robust security measures, continuous monitoring, and a proactive approach to mitigate risks and safeguard sensitive information in the digital era. An algorithm is developed to enhance the security of data during transition over the network.

1. Introduction

Cloud computing relies on efficient data storage, processing, and retrieval. The importance of data in cloud computing includes scalability, accessibility, cost efficiency, data security, data analytics etc. Security in cloud computing faces challenges such as data breaches, identity theft, and unauthorized access. Issues include shared data vulnerabilities, compliance concerns, and the need for robust encryption. Regular updates, secure APIs, and user education are essential to address these challenges. Encryption and decryption algorithms play a crucial role in safeguarding information. Encryption involves converting plaintext into unreadable ciphertext, while decryption reverses this process, ensuring that only authorized parties can access sensitive data. Common algorithms include like AES, DES, 3DES and Blowfish.

2. Problem Definition

The problem identified with security issues in the cloud lies in the potential vulnerabilities and risks that can compromise the confidentiality, integrity, and availability of data and services. The dynamic nature of cloud environments, involving shared resources, external access points, and complex configurations, introduces challenges that need careful consideration. These problems include:

- Uncertain Data Location and Jurisdiction
- Limited Control over Infrastructure
- Complexity of Shared Environments
- Rapid Changes and Updates
- Dependency on Service Providers
- Inadequate Security Posture
- Insufficient Monitoring and Logging
- Challenges in Compliance and Auditing
- Potential for Data Interception
- Emerging Threat Landscape

Addressing these problems requires a comprehensive approach, combining technical solutions, robust policies, ongoing training, and a collaborative effort between organizations and their cloud service providers to ensure a secure cloud computing environment.

3. Existing Algorithms

Security issues encompass a broad range of concerns in the digital realm. Regularly updating encryption methods is essential to stay ahead of evolving security threats. Encryption and decryption algorithms play a crucial role in ensuring the confidentiality, integrity, and authenticity of sensitive information. Encryption and decryption algorithms are fundamental to maintaining the security and privacy of digital information. They serve as a cornerstone in protecting data across various industries, securing communications, and ensuring compliance with legal and regulatory requirements. The existing encryption and decryption algorithms taken for study are

- **Data Encryption Standard (DES):** A widely used symmetric algorithm, though it is now considered insecure due to its short key length.
- **Triple DES (3DES):** A more secure variant of DES that applies the DES algorithm three times with different keys.
- **Advanced Encryption Standard (AES):** Currently one of the most widely used symmetric algorithms. AES supports key lengths of 128, 192, or 256 bits.
- **Blowfish:** A symmetric key block cipher known for its variable key length and speed.

3.1 Data Encryption Standard (DES)

The Data Encryption Standard (DES) is a symmetric-key algorithm for the encryption of electronic data. It is a block cipher algorithm with key length of 56 bit. The entire data is divided into 64 bit blocks which will be sent as the input. The output will be a 64 bit blocks of encrypted text known as cipher text.

3.1.1 Working Methodology

The Data Encryption Standard (DES) algorithm follows a Feistel network structure. Here's a simplified overview of the DES encryption methodology:

- **Initial Permutation (IP):** The 64-bit plaintext block undergoes an initial permutation.
- **Key Generation:**** A 56-bit key is used for encryption. This key undergoes a permutation and is divided into two 28-bit halves. This process is repeated for each of the 16 rounds.

- **Rounds (16 in total):**
  - **Expansion:** The right half (32 bits) is expanded to 48 bits.
  - **Key Mixing (XOR):** The expanded right half is XORed with the subkey for the current round.
  - **Substitution (S-boxes):** The 48-bit result is substituted using eight S-boxes, each providing a 4-bit output.
  - **Permutation (P-box):** The 32-bit output from the S-boxes undergoes a permutation.
  - **XOR with Left Half:** The output from the P-box is XORed with the left half of the data.
  - **Swap:** The left and right halves are swapped.
  - **Final Permutation (IP-1):** After 16 rounds, a final permutation is applied to the swapped left and right halves.

The result is the 64-bit ciphertext block. DES has been widely used, but its short key length (56 bits) makes it susceptible to brute-force attacks. As a result, more secure encryption algorithms like AES are now recommended for most applications.

3.1.2 Advantages of DES

- Standardization
- Speed
- Simplicity
- Feistel Structure

3.1.3 Disadvantages of DES

However, despite these advantages, the main disadvantage of DES is its key length. As computational power increased, the 56-bit key length became susceptible to brute-force attacks. Its key length is now considered insufficient for secure cryptographic applications. Organizations and individuals are strongly encouraged to use more modern and secure encryption algorithms.

3.2 Triple Data Encryption Standard (3DES)

Triple DES, also known as 3DES or TDEA (Triple Data Encryption Algorithm), is a symmetric key block cipher that applies the Data Encryption Standard (DES) algorithm three times to each data block. DES itself is a block cipher that operates on 64-bit blocks of data using a 56-bit key. However, due to advances in computing power and the vulnerability of DES to brute-force attacks, Triple DES was introduced to provide a higher level of security.

3.2.1 Methodology

Key Generation:

- Triple DES uses three 56-bit keys (K1, K2, K3), resulting in a total key length of 168 bits (56 bits * 3).
- The three keys can be independent or related, depending on the keying option chosen.

Encryption:

- The plaintext is initially encrypted with the first key (K1).
- Then, the ciphertext is decrypted with the second key (K2).
- Finally, the result is encrypted again with the third key (K3).

The encryption process can be represented as follows:

Ciphertext=EK3(DK2(EK1(Plaintext)))

Decryption:

- Decryption is the reverse process of encryption.
- The ciphertext is decrypted with the third key (K3).
- The result is encrypted with the second key (K2).
- Finally, the output is decrypted with the first key (K1).

The decryption process can be represented as follows:

Plaintext=DK1(EK2(DK3(Ciphertext)))

Triple DES provides a higher level of security than DES alone, but it's worth noting that its effective key length is reduced to 112 bits due to the meet-in-the-middle attack. As a result, more advanced encryption algorithms like AES (Advanced Encryption Standard) are often preferred for modern cryptographic applications.

3.2.2 Merits:

- **Security Improvement Over DES:** Triple DES enhances the security of the original DES algorithm by applying it three times consecutively, making it more resistant to brute-force attacks.
- **Backward Compatibility:** Triple DES is designed to be backward compatible with DES. Existing systems using DES can transition to Triple DES with minimal modifications.
- **Robustness:** The triple application of DES in Triple DES provides a higher level of cryptographic strength compared to DES alone, making it more resilient against various attacks.
- **Gradual Transition:** Organizations can transition from DES to Triple DES in a phased manner, upgrading systems gradually without requiring an immediate and complete overhaul.

3.2.3 Demerits (Disadvantages) of Triple DES:

- **Key Length and Security Concerns:** While Triple DES offers increased security compared to DES, its effective key length is reduced to 112 bits due to the meet-in-the-middle attack. Modern standards like AES offer equivalent or better security with longer effective key lengths.
- **Performance:** Triple DES is computationally more intensive than DES because it involves three passes of the DES algorithm. This can impact performance, especially in resource-constrained environments.
- **Vulnerability to Meet-in-the-Middle Attack:** Despite its triple application, Triple DES is still vulnerable to a meet-in-the-middle attack, which reduces its effective key length. This weakness is one reason why more modern symmetric encryption algorithms are preferred.
- **Complexity and Key Management:** Managing three keys for Triple DES can be more complex than managing a single key for other encryption algorithms. Key distribution and management become more challenging, especially in large-scale systems.

- **Limited Block Size:** Like DES, Triple DES has a fixed block size of 64 bits. In modern cryptography, larger block sizes are often preferred for certain applications, and Triple DES may not be suitable for these cases.

While Triple DES has served as a reliable encryption algorithm for many years, its disadvantages, such as its key length limitations and computational intensity, have led to its gradual replacement by more advanced encryption standards like AES in modern cryptographic applications.

3.3 Advanced Encryption Standard (AES)

AES, or the Advanced Encryption Standard, is a widely used symmetric encryption algorithm designed to provide a high level of security and efficiency. It was established as the encryption standard by the U.S. National Institute of Standards and Technology (NIST) in 2001, succeeding the Data Encryption Standard (DES). AES is a symmetric key algorithm, meaning the same key is used for both encryption and decryption.

3.3.1 Key features of AES include:

Key Sizes: AES supports key sizes of 128, 192, or 256 bits. The security strength of AES increases with the key size.

Block Size: AES operates on blocks of data, and it has a fixed block size of 128 bits.

3.3.2 Encryption Process:

- **Key Expansion:** The original key is expanded into a key schedule, creating a set of round keys.
- **Initial Round (AddRoundKey):** Each byte of the block is combined with the corresponding byte from the round key using bitwise XOR.
- **Rounds (9/11/13 rounds for 128/192/256-bit keys):** A series of rounds are performed, each consisting of four transformations: SubBytes, ShiftRows, MixColumns, and AddRoundKey.
- These transformations provide diffusion and confusion, making it difficult for an attacker to discern patterns.
- **Final Round:** The final round excludes the MixColumns transformation.
- **Decryption Process:**
- **Key Expansion:** Similar to encryption, the original key is expanded into a key schedule.
- **Inverse Rounds:** The decryption process involves performing the inverse of each encryption round in reverse order: InvAddRoundKey, InvMixColumns, InvShiftRows, and InvSubBytes.
- **Final Round:** Similar to the encryption process, the final round excludes the InvMixColumns transformation.
- **Security:** AES has withstood extensive cryptanalysis and is considered secure when used with an appropriate key size. The security of AES is based on the key length, and it is widely used in various applications, including securing communications over the Internet, encrypting files, and protecting sensitive data.
- **Performance:** AES is designed for efficiency and is computationally fast on a wide range of devices, from resource-constrained embedded systems to high-performance servers. AES is a widely adopted symmetric encryption algorithm known for its security,

efficiency, and versatility. It has become the standard for symmetric key encryption in a broad range of applications.

3.3.3 Merits:

- **Security:** AES is widely recognized as a secure encryption algorithm. It has withstood extensive cryptanalysis and is considered resistant to various types of attacks.
- **Key Length Options:** AES supports key sizes of 128, 192, and 256 bits, providing flexibility for users to choose a level of security that meets their specific needs.
- **Efficiency:** AES is designed to be computationally efficient, making it suitable for a wide range of devices and applications, from resource-constrained embedded systems to high-performance servers.
- **Standardization:** AES is a widely adopted and standardized encryption algorithm, ensuring interoperability and compatibility across different systems and platforms.
- **Versatility:** AES can be used for various cryptographic applications, including securing communications over the Internet, encrypting files, and protecting sensitive data in a variety of contexts.
- **Well-Defined Structure:** The algorithm has a well-defined and straightforward structure, making it easy to implement and integrate into different software and hardware systems.

3.3.4 Demerits:

- **Key Management:** Like any symmetric encryption algorithm, key management can be challenging, especially in large-scale systems. Safely distributing and storing secret keys is crucial for maintaining security.
- **Fixed Block Size:** AES has a fixed block size of 128 bits. While this is sufficient for many applications, some modern cryptographic scenarios benefit from larger block sizes.
- **Potential Side-Channel Attacks:** In certain environments, side-channel attacks like timing or power analysis could potentially be used to gain information about the encryption key. Implementations need to consider and guard against such attacks.
- **Not Quantum Resistant:** While AES is secure against classical computing attacks, it is not quantum-resistant. The advent of quantum computers could potentially compromise the security of AES. Post-quantum cryptography solutions are being explored for long-term security.
- **Limited to Symmetric Key Encryption:** AES is a symmetric key encryption algorithm, meaning the same key is used for both encryption and decryption. This limits its applicability in scenarios where a public-key infrastructure is desired. AES is a highly secure and efficient encryption algorithm with widespread adoption. Its disadvantages are relatively minor in comparison to its advantages, and it remains a cornerstone in modern cryptographic practices.

3.4 BlowFish Algorithm

Blowfish is a symmetric-key block cipher that was designed by Bruce Schneier in 1993. It was designed as a fast, free alternative to existing encryption algorithms like DES (Data Encryption Standard) and IDEA (International Data Encryption Algorithm). Blowfish is known for its simplicity, speed, and security.

3.4.1 Key features:

- **Symmetric-Key Algorithm:** Blowfish is a symmetric-key algorithm, which means that the same key is used for both encryption and decryption.
- **Block Cipher:** Blowfish operates on fixed-size blocks of data, typically 64 bits. The algorithm divides the input data into blocks and processes each block independently.
- **Variable Key Length:** One notable feature of Blowfish is its variable key length. The key can be any length from 32 bits to 448 bits, making it adaptable for different security needs. Longer keys generally provide stronger security.
- **Substitution-Permutation Network (SPN):** Blowfish uses a substitution-permutation network structure, where a series of substitutions and permutations are applied to the input data in multiple rounds. The number of rounds is a parameter, and Blowfish allows for a variable number of rounds, usually ranging from 16 to 20.
- **Feistel Network:** Blowfish employs a Feistel network structure, which is a specific type of symmetric structure used in block cipher design. In a Feistel network, the data block is divided into two halves, and each half is processed separately through a series of functions. The two halves are then combined in some way.
- **Key Expansion:** Blowfish uses a key expansion process to generate a series of subkeys from the original key. These subkeys are then used in the encryption and decryption processes.
- **Security:** Blowfish was designed with a focus on security and has withstood extensive cryptanalysis over the years. However, since it is an older algorithm, its usage may be limited in certain applications due to the availability of more modern alternatives.

While Blowfish has been widely used and studied, it is worth noting that newer algorithms such as AES (Advanced Encryption Standard) are more commonly recommended for secure communication today, as they have undergone more extensive scrutiny and are considered more robust against certain types of attacks. Blowfish, like any encryption algorithm, has its merits and demerits. Here are some of the advantages and disadvantages of the Blowfish encryption algorithm:

3.4.1 Merits:

- **Variable Key Length:** Blowfish allows for a variable key length, which provides flexibility in choosing the level of security. Users can select key lengths that suit their specific requirements.
- **Fast Encryption Speed:** Blowfish is known for its fast encryption and decryption speed, making it suitable for applications where performance is a critical factor.
- **Simple and Elegant Design:** The algorithm's design is relatively simple and elegant, making it easier to implement and understand. This simplicity can be advantageous for certain applications.
- **Well-Studied:** Blowfish has been extensively studied by the cryptographic community since its introduction. It has withstood a significant amount of cryptanalysis, and no practical vulnerabilities have been discovered to date.

- **No Known Backdoors:** As of my last knowledge update in January 2022, there are no known backdoors in the Blowfish algorithm. This adds to its credibility and trustworthiness.

3.4.2 Demerits:

- **Limited Key Lengths for Some Applications:** While the variable key length is an advantage, for certain high-security applications, the maximum key length of 448 bits in Blowfish may be considered insufficient compared to some modern algorithms.
- No Longer Considered State-of-the-Art: Blowfish was designed in the early 1990s, and since then, more advanced encryption algorithms like AES have been developed and widely adopted. AES is generally considered more secure and is the current state-of-the-art symmetric-key encryption standard.
- **Lack of Formal Standardization:** Blowfish has not undergone the same level of formal standardization as algorithms like AES. This can be a drawback in situations where adherence to established standards is crucial.
- **Potential Vulnerabilities to Future Attacks:** As cryptographic techniques and computational power advance, older algorithms like Blowfish may become more susceptible to new types of attacks. While no practical vulnerabilities are known, the lack of ongoing development and standardization may limit its ability to adapt to future threats.
- **Not Suitable for Some Government Applications:** Blowfish is not approved for government use in the United States. Government agencies often have specific encryption standards, and Blowfish may not meet those requirements.

Blowfish has some favourable characteristics such as speed and simplicity, it's essential to consider the specific requirements and security standards of the application. For many modern applications, newer algorithms like AES are generally preferred.

4. The ENCIPHERMENT Algorithm

The algorithms discussed above has their own merits and demerits. To overcome the demerits and to enhance the security a new algorithm ENCIPHERMENT has been fostered. It is a symmetric algorithm. It uses block cipher technology, where the data is split into 64 bit blocks. Furthermore each block is sub divided into 8 blocks, each block contains 8 bits. The sub blocks are named as B1, B2,…..,B8. The Logical XOR Operation is performed B1 and B2 to compute R1, similarly XOR operation is performed on (B3,B4), (B5,B6) and (B7,B8) to compute R2, R3 and R4 respectively. Next, the blocks B1, B3, B5 and B7 are divided into equal halves which are termed as B1L, B1R, B3L, B3R, B5L, B5R and B7L, B7R respectively. In the next step, the above blocks are negated using logical NOT operation and we can obtain NB1L, NB1R, NB3L, NB3R, NB5L, NB5R and NB7L, NB7R respectively. After negating the blocks, NB1L and NB1R, NB3L and NB3R, NB5L and NB5R and NB7L and NB7R are swapped to obtain RE1, RE2, RE3 and RE4 respectively. Finally, R1,R2,R3, R4 and RE1, RE2, RE3, RE4 are the cypher text. Therefore encryption process has been completed and the encrypted text will be transmitted over the network in cloud environment.

4.1.1 Decryption Process

The values of RE1, RE2, RE3 and RE4 are negated and swapped to retrieve the values of B1, B3, B5 and B7 respectively. Next, XOR operation is performed on (R1, B1), (R2, B3), (R3, B5) and (R4, B7) to obtain the plain text B2, B4, B6 and B8.

Therefore the bits are encrypted and the cipher text are sent to the receiver over the cloud network. Once the cipher text reaches the receiver, it is been decrypted and the original plain text will be generated from the encrypted data. I expect that this algorithm will be more efficient and reduce the complexity of encrypting and decrypting the data over the cloud environment. It also increases the security level so the reliability will also gradually increase.

4.2 Pseudo Code

```
1. Start
2. Read the plain text and store in text
3. Split text into 64 bits and store in t1, t2,t3…….. tn.
4. Assign s=1
5. Repeat steps 6 to 7 till s=tn
6. Assign B[s]=8bits
7. S++
8. ComputeR1=B[1] XOR B[2]
9. ComputeR2=B[3] XOR B[4]
10. ComputeR3=B[5] XOR B[6]
11. ComputeR4=B[7] XOR B[8]
12. Repeat steps 13 to 15 for s=1 till s=4
13. BL[s]=first 4 bits of B[s]
14. BR[s]0last 4 bits of B[s]
15. Assign s=s+2
16. Repeat steps 17 to 18 for s=1 till s=4
17. NBL[s]=NOT (BL[s])
18. Assign s=s+2
19. Repeat steps 20 to 18 for s=1 till s=4
20. Temp=NBL[s]
21. NBL[s]=NBR[s]
22. NBR[s]=Temp
23. RE[a]=NBL[s] + NBR[s]
24. Assign s=s+2
25. R1,R2,R3,R4,RE[1],RE[2],RE[3] & RE[4] are cipher text
```

6. Conclusion

In conclusion, encryption and decryption techniques play a fundamental role in ensuring the confidentiality, integrity, and security of sensitive information in various contexts. As technology advances and the need for secure communication grows, these techniques continue to evolve to meet the challenges posed by potential threats and vulnerabilities. Encryption serves as a robust method for transforming plaintext into ciphertext, rendering it unintelligible to unauthorized users. Through the use of cryptographic algorithms and keys, encryption provides a powerful tool for protecting data both at rest and in transit. Whether it's securing personal communications, financial transactions, or sensitive government information, encryption forms the backbone of modern cybersecurity. Decryption, the reverse process of encryption, allows authorized parties to transform ciphertext back into its original plaintext form. Access to the decryption key is crucial for this process, ensuring that only those with the proper authorization can decipher and comprehend the protected information. The balance between the strength of encryption algorithms and the need for efficient processing is a continuous challenge. Striking the right balance is essential to maintain both security and performance. As cryptographic techniques advance, considerations for post-quantum cryptography become increasingly important to address the potential threats posed by quantum computers. In the realm of privacy and data protection, encryption stands as a cornerstone, enabling individuals and organizations to assert control over their information. However, it's important to note that while encryption is a powerful safeguard, its effectiveness depends on proper implementation, key management, and adherence to security best practices. Encryption and decryption techniques are indispensable tools in the ongoing efforts to secure digital communication and protect sensitive data. As the digital landscape evolves, the continued refinement and advancement of these techniques will be crucial for maintaining the privacy and security of information in our interconnected world. This Enchiperment algorithm may increase the efficiency of security and it may be reliable when compared to the existing algorithms.

References

1. A Survey of Attacks Security Mechanisms and Challenges in Wireless Sensor Networks, G. Padmavathi and D. Shanmugapriya, (IJCSIS'09) International Journal of Computer Science and Information Security,vol. 4 no. 1&2 2009.
2. Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities and Countermeasures, Y. Kumar R. Munjal and H. Sharma "”, International Journal of Computer Science and Management Studies, vol. 11 no. 3 Oct 2011.
3. Data Security and Privacy Protection Issues in Cloud Computing, **International Conference on Computer Science and Electronics Engineering,** March 2012
4. Comparative Study on Data Encryption Algorithms in Cloud Platform, D.Palanivel Rajan, Dr. S. John Alexis, International Journal of Engineering Research & Technology (IJERT), Vol. 6 Issue 10, October – 2012
5. A true random-based differential power analysis countermeasure circuit for an AES engine, P. Liu H. Chang and C. Lee, IEEE TRANSACTIONS on CIRCUITS and SYSTEMS-II: EXPRESS BRIEFS,vol. 59 no. 2 pp. 103-107 2012.
6. A Study of Encryption Algorithms AES DES and RSA for Security, P. Mahajan and A. Sachdeva , Global Journal of Computer Science and Technology Network Web & Security,vol. 13 no. 15 2013.
7. A study and analysis on symmetric cryptography, S. Chandra, S. Bhattacharyya, S. Paira and S. S. Alam, Science Engineering and Management Research (ICSEMR), 2014 International Conference on, Chennai, 2014, pp. 1-8.
8. A Survey on Encryption Schemes in Wireless Sensor Networks, H. Hayouni, M. Hamdi and T. H. Kim, Advanced Software Engineering and Its Applications (ASEA), 2014 7th International Conference on, Haikou, 2014, pp. 39-43.

ISSN:2153-182X, E-ISSN: 2153-1838
Vol. 17 No. 02 (2023)

9. Challenges and security issues in cloud computing from two perspectives: Data security and privacy protection, S. Mahdi Shariati; Abouzarjomehri; M. Hossein Ahmadzadegan, International Conference on Knowledge-Based Engineering and Innovation (KBEI), 2015
10. Exploring Data Security Issues and Solutions in Cloud Computing, Ravi Kumar, P.Herbert Raj, Procedia Computer Science, Volume 125, 2018.